In the Claims:

1. (Currently Amended) A method, at least partially implemented by a computing device, for processing a database query, comprising:

partially pre-aggregating records in a database ~~according to a single grouping column~~ to provide a result that contains at least two records having like grouping column values;

aggregating records derived from the partial pre-aggregation to provide a result that contains records having unique grouping column values; and

~~estimating the costs and benefits of the partial pre-aggregation, and~~ partially pre-aggregating the records only if ~~the estimating indicates that the benefits are greater than the costs.~~ an estimation, based in part on a calculation of a probability that a record will be absorbed by a group of records already in memory, indicates that a number of output records from the partial pre-aggregation will be significantly less than a number of input records to the partial pre-aggregation.

2. (Original) The method as recited in claim 1, wherein the partially pre-aggregating further comprises:

maintaining a record store in memory, the record store having one record for each different grouping column value encountered in the operation;

receiving a new record;

combining the new record with a record having the same grouping column value, if such a record exists; and

adding the new record to the record store in the memory if there is no record in the record store that has the same grouping column value as the new record.

3. (Original) The method as recited in claim 2, further comprising:

adding additional new records to the record store until the record store reaches a capacity such that it can accept no new records; and

outputting one or more records from the record store to a subsequent database operator.

4. (Original) The method as recited in claim 3, wherein after the one or more records have been output to the subsequent database operator, the adding and outputting are repeated until there are no new records to process.

5. (Original) The method as recited in claim 4, wherein any records remaining in the record store after there are no new records to process are output to the subsequent database operator.

6. (Original) The method as recited in claim 3, wherein the subsequent database operator is a join.

7. (Cancelled)

8. (Original) The method as recited in claim 1, wherein the partially pre-aggregating includes utilizing a hashing function.

9. (Original) The method as recited in claim 1, wherein the partial pre-aggregating creates a record store in memory, and wherein the method further comprises utilizing the record store in memory for one or more other database operators.

10. (Cancelled)

11. (Original) A computer programmed to perform the method recited in claim 1.

12—23. (Cancelled)

24. (Currently Amended) A relational database computer program stored on a computer-readable medium, the relational database computer program comprising computer-executable instructions that, when executed on a computer, perform the following steps:

receiving a stream of input records;

partially pre-aggregating the input records according to a single grouping column to provide a result that contains at least two records having like grouping column values, wherein the partial pre-aggregation is performed if an estimation, based in part on a calculation of a probability that a record will be absorbed by a group of records already in memory, indicates that a number of output records from the partial pre-aggregation will be significantly less than a number of input records to the partial pre-aggregation;

joining the partially pre-aggregated records with other data to create a record store; and

aggregating records within the record store to provide a result that contains records having unique grouping column values.

25. (Original) The relational database computer program as recited in claim 24, wherein:

the record store has a capacity that is less than the number of records in the stream of input records; and

the aggregating each input record is performed until the record store reaches capacity.

26. (Cancelled)

27. (New) The method of claim 1, wherein the estimation is based, in part, on an estimated absorption rate by which records are absorbed by records in memory.

28. (New) The method of claim 27, wherein the absorption rate of available memory is estimated, in part, based on a number of records expected to be processed.

29. (New) The method of claim 28, wherein the number of records expected to be processed is estimated, in part, based on a number of records that will fit in memory.

30. (New) The method of claim 1, wherein the estimation is based, in part, on factors comprising:

the number of output records, T(N);

the number of input records, N; and

the relationship:

$$T(N) = M + (N - M)(1 - A(R(M))) = M + (N - M)\sum_{i=1}^{D}(1 - p_i)^{R(M)};$$

wherein M records can fit in memory.

31. (New) The method of claim 30, wherein the number of input records is known.

32. (New) The method of claim 30, wherein the number of input records is estimated.